Course guide
340376 - INEP-I3O23 - Introduction to Software Engineering

Last modified: 12/06/2024

Unit in charge: Vilanova i la Geltrú School of Engineering
Teaching unit: 747 - ESSI - Department of Service and Information System Engineering.

Degree: BACHELOR'S DEGREE IN INFORMATICS ENGINEERING (Syllabus 2018). (Compulsory subject).

Academic year: 2024 **ECTS Credits:** 6.0 **Languages:** Catalan

LECTURER

Coordinating lecturer: Franch Gutiérrez, Xavier

Others: López Cuesta, Lídia

PRIOR SKILLS

Basic programming knowledge

REQUIREMENTS

Programming methodologies

DEGREE COMPETENCES TO WHICH THE SUBJECT CONTRIBUTES

Specific:
3. CEFC1. Ability to design, develop, select and value applications and informatic systems affirming its reliability, security and quality corresponding to ethical principals and legislation and current rules.
6. CEFC2. Ability to plan, conceive, develop, manage informatic projects, services and systems in all areas, leading their implementation and continuous improvement assessing their economic and social repercussions.
1. CEFC16. Knowledge and application of principals, methodologies, life cycles of programming engineering.
2. CEFC6. Basic knowledge and application of algorithmic processes, informatic techniques to design solutions of problems, analyzing if proposed algorisms are apt and complex.
8. CEFC8. Ability to analyze, to design, to construct and to maintain applications in a well built, secure and efficient way choosing the most adequated paradigms and languages.
10. CETI5. Ability to select, to develop, integrate and manage information systems which satisfy organization necessities with indentified costs and quality criteria.
11. CEFC3. Ability to understand the importance of negotiation, effective work habits, leadership and communication skills in all environments for software development.

Transversal:
9. SELF-DIRECTED LEARNING - Level 2: Completing set tasks based on the guidelines set by lecturers. Devoting the time needed to complete each task, including personal contributions and expanding on the recommended information sources.
12. SELF-DIRECTED LEARNING. Detecting gaps in one's knowledge and overcoming them through critical self-appraisal. Choosing the best path for broadening one's knowledge.
16. EFFICIENT ORAL AND WRITTEN COMMUNICATION. Communicating verbally and in writing about learning outcomes, thought-building and decision-making. Taking part in debates about issues related to the own field of specialization.
18. TEAMWORK - Level 2. Contributing to the consolidation of a team by planning targets and working efficiently to favor communication, task assignment and cohesion.
20. TEAMWORK. Being able to work as a team player, either as a member or as a leader. Contributing to projects pragmatically and responsibly, by reaching commitments in accordance to the resources that are available.

TEACHING METHODOLOGY

A case study that conveys the course.

Material in the form of slides, for study and review of contents.

Exercises and problems that students must solve on their own as a means of understanding the contents of the subject.

A project in groups of three people as a fundamental mechanism for learning and evaluation.

A number of activities during the course, designed as a mechanism for learning and evaluation.

Tutoring lab sessions of work done.

LEARNING OBJECTIVES OF THE SUBJECT

Understand the concept and importance of software engineering.

Know the various stages of the development of a software system, and the different objectives of each stage.

Be able to understand a complex problem, and to model it.

Know how to document a specification.

Be able to design the architecture of a system that meets the needs of the specified problem.

Be able to design a relational database, create it and make basic queries with the SQL language

Know how to work in a team.

STUDY LOAD

Type	Hours	Percentage
Hours large group	30,0	20.00
Self study	90,0	60.00
Hours small group	30,0	20.00

Total learning time: 150 h

CONTENTS

Introduction to software engineering (SE)

Description:
It presents the basic concepts of SE that will be developed throughout the course

Specific objectives:
Know the need and understand the concept of SE.
Know the concept of the software life cycle as a combination of several stages.
Know the stages of the life cycle.

Related activities:
C1

Full-or-part-time: 2h
Theory classes: 2h

Software system specification

Description:
It presents the requirements engineering stage and the various models that allow the requirements to be described rigorously

Specific objectives:
Know the specific activities of the requirements engineering stage.
Know and apply the models provided by the UML language for the specification of functional requirements.
Know and apply systematic approaches in the description of non-functional requirements.

Related activities:
C1, Pr, T1

Full-or-part-time: 34h
Theory classes: 12h
Laboratory classes: 10h
Self study : 12h

Architectural design of software systems

Description:
It presents the software design stage, based on the application of patterns

Specific objectives:
Know the activities specific to the software design stage.
Know the concept of pattern, and the distinction between architectural pattern and design pattern.
Know some basic patterns of both architecture and design, and understand their effects on requirements.
Learn to transform a problem specification into a system design, through the systematic application of patterns.

Related activities:
C2, Pr, T2

Full-or-part-time: 22h
Theory classes: 8h
Laboratory classes: 6h
Self study : 8h

Introduction to persistence and data bases

Description:
Introduces the essential aspects of data persistence based on relational technology

Specific objectives:
Understand the concept of data persistence.
Understand relational data representation technology.
Learn how to transform a data specification expressed in a UML conceptual model, into a relational model.
Learn how to make simple queries on relational models using the SQL language.

Related activities:
C2, Pr, T3

Full-or-part-time: 18h
Theory classes: 6h
Laboratory classes: 6h
Self study : 6h

Conclusion

Description:
It summarizes the knowledge and practices taught in the subject and advances the challenges that will be studied in the AMEP subject

Specific objectives:
Summarize and relate the knowledge and practices taught in the subject.
List the limitations of the techniques seen.
Advance some solutions to these limitations, and possible alternative techniques

Related activities:
T4

Full-or-part-time: 2h
Theory classes: 2h

Development of a practice

Description:
Apply the knowledge introduced during the course in a laboratory practice developed as a team

Specific objectives:
Learn to effectively use a software development environment appropriate for the course objectives.
Learn to analyze the statement of a non-trivial problem for its implementation.
Apply the theoretical notions of specification, design and persistence in the use case defined in the practice statement.
Face the typical challenges of programming a system by applying the principles and methods of software engineering.

Related activities:
Pr, C2

Full-or-part-time: 72h
Laboratory classes: 8h
Self study : 64h

GRADING SYSTEM

During the course, there will be 9 assessment events: 2 tests that will be carried out during the exam periods (C1, C2), 4 tasks/questionnaires associated with the main topics of the subject, which will be done in theory class (Tx), and the practice (Pr) whose grade will be calculated from 3 deliveries.

Controls are re-evaluable in the Re-evaluation Period (each control is re-evaluated separately).

EXAMINATION RULES.

See Catalan version

BIBLIOGRAPHY

Basic:
- Pressman, Roger S. Ingeniería del software : un enfoque práctico [on line]. 9a ed. México [etc.]: McGraw-Hill, 2021 [Consultation: 16/02/2024]. Available on: https://www-ingebook-com.recursos.biblioteca.upc.edu/ib/NPcd/IB_BooksVis?cod_primaria=1000187&codigo_libro=4272. ISBN 9781456287726.
- Sommerville, Ian. Software engineering [on line]. 10th ed. Harlow: Pearson Education Limited, 2016 [Consultation: 14/02/2024]. Available on: https://ebookcentral-proquest-com.recursos.biblioteca.upc.edu/lib/upcatalunya-ebooks/detail.action?pq-origsite=primo&docID=5185655. ISBN 9781292096148.
- Garcia-Molina, Hector; Ullman, Jeffrey D; Widom, Jennifer. Database systems : the complete book [on line]. 2nd ed. Harlow, Essex: Pearson Education Limited, 2013 [Consultation: 14/02/2024]. Available on: https://ebookcentral-proquest-com.recursos.biblioteca.upc.edu/lib/upcatalunya-ebooks/detail.action?pq-origsite=primo&docID=5174436. ISBN 9781292037301.
- Larman, Craig. Applying UML and patterns : an introduction to object-oriented analysis and design and iterative development. 3th ed. Upper Saddle River, N.J.: Prentice Hall PTR, 2005. ISBN 0131489062.